A TEXT-TO-PICTURE SYSTEM FOR RUSSIAN LANGUAGE

Dmitry Ustalov
Ural Federal University
e-mail: dmitry@eveel.ru

Abstract

This paper presents motivation and design of the general purpose text-to-picture synthesis system. The described TTP system is designed for Russian language processing and operates with the natural language analysis subsystem, the stage processing subsystem, and the rendering subsystem. Every processing stage has been described and the basic design ideas of the system architecture have been highlighted. User study has been performed and further work reasons are explained.

Keywords: *text-to-picture synthesis, text-to-scene synthesis, natural language processing, computational linguistics, information visualization, semantic representation.*

1. INTRODUCTION

A picture is worth of a thousand words. The text-to-picture synthesis problem is important because there are many domains exist where clearness of textual information is necessary: foreign language learning [1], traffic accident visualization [2], rehabilitation of people with cerebral injuries [3], etc.

2. RELATED WORKS

There are several fully-functional analogues that are described in various papers. These systems can be classified into two categories:

1. *General purpose systems* which perform visualization of the unrestricted natural language text aimed to convey the meaning of that text;
2. *Problem-oriented systems* that have been designed to operate with restricted subset of natural language in terms of the specified domain. These systems have often been meant to be used by graphics designers as an alternative way to specify the layout of a scene.

2.1 General Purpose TTP Systems

There are two notable general purpose TTP systems: Word2Image [4] and the TTP project of University of Wisconsin [3, 5].

The Word2Image system generates picture collages based on annotated photo albums from the popular Flickr website. Collages are composed from photos that correspond to the keywords of the input text.

The TTP project of University of Wisconsin aims to convey the meaning of the English text by revealing the important objects and their relations.

2.2 Problem-Oriented TTP Systems

There are also four notable problem-oriented TTP systems: WordsEye [6], SPRINT [7], NALIG [8], and CarSim [2].

The WordsEye system is designed to operate with 3D images in mostly unrestricted subset of the English language aimed to spatial attributes of actors — the interacted objects of the natural language text. This system uses a statistical natural language parser, a set of depiction rules in the S-expression form, a proprietary 3D animation system, and 3D models from the Viewpoint model gallery.

Such systems as SPRINT and NALIG produce spatial reasoning visualization of the simple descriptive sentences. SPRINT operates with the Japanese language, and NALIG operates with the English language.

The CarSim system converts special-domain narratives on road accidents into an animated scene using icons.

3. MOTIVATION

It is estimated that 60% children in Russian Federation have various speech impairments [9] that result in them relying on techniques other than natural speech alone for communication.

Unfortunately, it is impossible to find a TTP system that is able to work with the Russian language because of the processing complexity, the lack of available dictionaries, the corpora, and the necessary software.

Moreover, all existing TTP systems are either unavailable, discontinued, or have a proprietary license that makes it impossible to add the Russian language support to them.

4. A TTP SYSTEM FOR THE RUSSIAN LANGUAGE

It has been established that TTP systems have three stages of processing [6]:

1. A stage of *linguistic analysis* — tokenization, morphological and syntactic parsing, obtaining semantic representation of the input text;
2. A stage of *depictors generation* — generation of the set of graphical depictors that correspond with the obtained semantic representation;
3. A final stage of *picture synthesis* — construction of a vector or a raster image from the graphical primitives that are positioned agreeing with generated depictors.

All the processing stages are presented at Figure 1: the Analyzer block represents the linguistic analysis stage, the Stage block represents the depictors' generation stage, and the Renderer block represents the picture synthesis stage.

In TTP systems, every processing stage strongly depends on many information resources, including thesauri, graphical primitives, depiction rules, and semantic descriptions of actors [10].

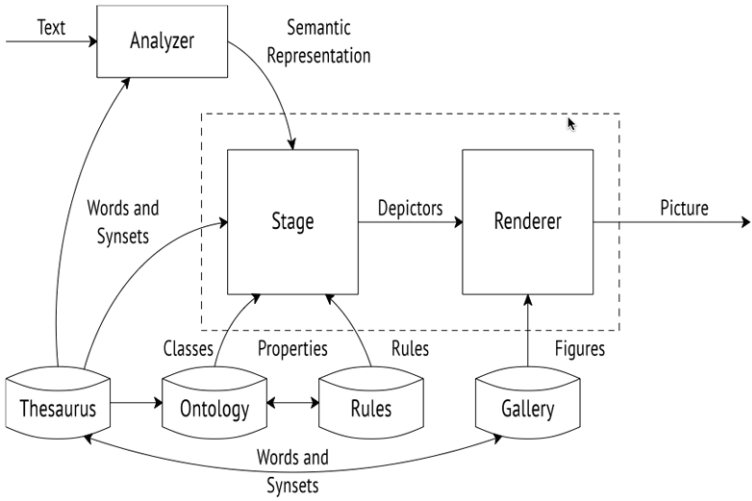

Figure 1. Text-to-Picture synthesis stages

4.1. Linguistic Analysis

Before the final picture has been rendered, it is required to perform some kind of a shallow semantic parsing of the input text. This process depends on two preliminary tasks: text tokenization and morphological annotation.

4.1.1. Tokenization

Tokenization is the first part of the linguistic analysis stage. Text should be split into paragraphs, sentences, sub-sentences and such individual tokens as words, digits, etc.

Greeb[1] is a simple heuristic tokenizer that is implemented in terms of a finite state machine. The state diagram of the FSM is presented at Figure 2.

Input alphabet of FSM is a set of Russian letters in UTF-8 encoding, Arabic digits, separators (e.g. space character), in-sentence punctuation marks (e.g. comma, dash, etc), punctuation marks (e.g. period, question sign) and End-of-Line/End-of-File signs.

The result of tokenization is a list of paragraphs $T=\{P_1,...,P_q\}$, where paragraph $P_i=\{S_1,...,S_p\}$ is a list of sentences, sentence $S_j=\{s_1,...,s_n\}$ is a list of subsentences, subsentence $s_k=\{t1,...,t_m\}$ is a list of the extracted tokens.

The advantage of using the described FSM is a relative simplicity of the high performance tokenizer implementation. However, the chosen method has some shortcomings, including the impossibility to process

[1] https://github.com/eveel/greeb

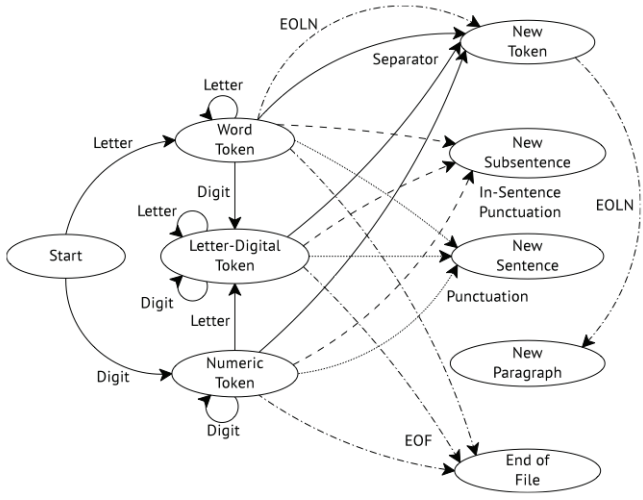

Figure 2. State diagram of FSM

texts with punctuation errors or texts with abbreviations. The shortcomings can be overcome using machine learning methods to identify the tokens of the input text, combined with the corpora and the thesauri to examine multiword tokens as single entities.

4.1.2. Morphological Analysis

It is required to perform the morphological analysis of the input text words, i.e. obtain the lemma, the POS (part of speech) tag, and the set of grammatical descriptors for each word in the text.

The Myaso[2] analyzer [11] is an open source dictionary-based morphological analysis framework that is designed for Russian language processing. This analyzer performs the POS tagging task as well as the lemmatization task.

Parsing can be performed by using the obtained morphological interpretation of words.

4.1.3. Parsing

Link Grammar for Russian is known as effective approach to performing the Russian language syntactic analysis [12]. Syntactic analysis requires tokenized and morphologically annotated text. Figure 3 demonstrates the link grammar of a translation into Russian of a sentence "Valery, your time has come". The extracted words can be mapped into the actors of the text.

[2] https://github.com/eveel/myaso

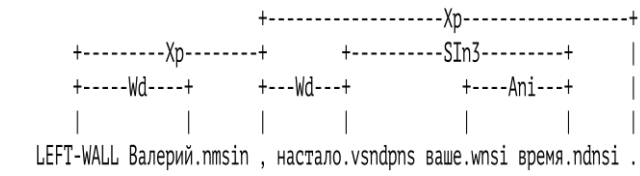

Figure 3. Link Grammar for Russian

Also, the extracted links can be obviously used to discover actions or attributes of these actors.

For example, the *A* link class corresponds to a noun phrase and can be treated as some attribute of an actor (e.g. a *pink pony*). The *S* link class indicates the verb phrase that can be found as unary expression — action of an actor (e.g. a *man has fallen* → *fall*(*men*)).

It should be also mentioned that some links are transitive that makes it possible to retrieve binary expressions — actions of two actors (e.g. *a man has fallen into the fire* → *fall*(*men*, *fire*)).

The result of the linguistic analysis is called the semantic representation of the input text. The semantic representation is an ordered pair $SR=(A,P)$ where A is a set of actors, while P is a set of n-ary verb expressions with these actors.

4.2. Depictors Generation

When semantic representation is built, it is necessary to transform the obtained actors and expressions into the depictors. Depictor is a simple instruction to Renderer which describes how the correspondent expression should be depicted in the final image.

The approach to depictors' generation is presented in [10]. Depictors have to be linked to specially defined depiction rules [6] that translate input expressions into the appropriate depictors. Depiction rules are found by lemmatized verbs from the expressions and map the expression arguments by defined rule operations.

For example, the *fall*(*men*, *fire*) expression would be translated into the following depictors using the `fallTo` depiction rule from [10]:

```
[
  [:rotate,   man, fire],
  [:together, man, fire]
]
```

Depictors are further executed by Renderer to make actors interaction more intuitive from the end user's point of view.

Figure 4. A "factory" icon from The Noun Project

4.3. Picture Synthesis

Actors' graphical primitives are critical to perform the final rendering. Due to the copyright reasons and the lack of necessary annotation, only few image libraries are suitable to be used in Russian language-oriented TTP system.

The Noun Project[3] is a large collection of icon images (called *nouns*) that are available under public domain or Creative Commons licenses. Thus, these images can be used for virtually any purpose. Icons of The Noun Project are annotated with tags, and some of those tags are written in Russian. This makes it possible to use The Noun Project as an image source for TTP system. An example of an icon from The Noun Project is presented at Figure 4.

4.3.1. Depictors Execution

The rendering subsystem uses given depictors to initialize the set of graphical primitives that are linked to the actors of the input text. Then, the renderer starts to resolve the primitives' mutual associations with the use of given depictors. It should be noted that the renderer is working with an assumption that every actor can be depicted using at least one graphical primitive.

The renderer iterates across the depictors list and executes each depictor of the list. Execution of depictor means performing predefined actions with the specified actor instances.

This means that the `[:rotate, man, fire]` depictor forces the renderer to rotate the *man* actor to the side of the *fire* actor, and the `[:together, man, fire]` depictor recommends the renderer to put these two actors together in the final image.

These instructions and recommendations are executed once and are stored in the renderer state, which is necessary for performing the picture layout task.

4.3.2. Picture Layout

The problem of finding the best positions for all the images is formulated at [5] as an optimization problem that can be solved by the Monte Carlo randomized algorithm. Nevertheless, the presented TTP system does not perform the keywords ranking, using the depictors instead.

[3] http://thenounproject.com

Thus, the picture layout computation problem can be formulated in the following way:where λ_k are weights, $o(I_i,I_j)$ is the area of an overlap

$$\lambda_1 \sum_{i=1}^{k} \sum_{j<i}^{k} \frac{o(I_i, I_j)}{A} + \lambda_2 \sum_{i=1}^{k} d(I_i) + \lambda_3 \sum_{i=1}^{k} \sum_{j<i}^{k} q(i,j) \to \min$$

between pictures I_i and I_j, A is the sum of the areas of all images, $d(I_i)$ is the distance of image I_i from the center of the picture, and $q(i,j)$ is the indicator function defined as4.3.3. Rendering

$$q(i,j) = \begin{cases} 1, & \textit{if there are : together depictor between actors of } i \textit{ and } j, \\ 0, & \textit{if not.} \end{cases}$$

When picture layout is computed, the renderer performs the last step of the TTP system operation. It creates the output file and places the graphical primitives in the computed place and state according to the layout generation results.

5. EXAMPLES

As an example, there are four images that been generated by the Utkus[4] system. Utkus is a general purpose TTP system that is designed to be suitable to perform visualization of short Russian texts, such as fragments of microblog posts, news summaries, comments on websites, literature for children, etc. These images are presented at Figures 5(a–d) and corresponding to the following texts:

1. A man has fallen into the fire[5];
2. Several houses[6];
3. There are a man and a woman in the house[7];
4. A certificate, a bear, a rain[8].

Due to the lack of space, the presented images have been cropped.

It should be noted that the Utkus system is unable to represent numerals (Figure 5(b)) at the present moment.

a) b) c) d)

Figure 5. Depiction of the texts: a) A man has fallen into the fire; b) Several houses; c) A certificate, a bear, a rain; d) There are a man and a woman in the house.

[4] http://utkus.eveel.ru

[5] Человек упал в огонь, in Russian.

[6] Несколько домов, in Russian.

[7] В доме находились мужчина и женщина, in Russian.

[8] Аттестат, медведь, дождь, in Russian.

6. USER STUDY

The user study highlights the system's usability and performance in discovering diversity and representativeness. The method of the TTP system user study is similar to [4]. Fifteen volunteering students have been invited to take part in the evaluation during the "Science-Art" exposition project in Yekaterinburg. The volunteers have been required to submit five short texts to the system and explore the representative images for each text. They have then been asked to fill in an assessment form including three questions as shown in Table 1. Each question requires a numerical answer based on the scale of: 1–strongly disagree, 2–disagree, 3–neutral, 4–agree, 5–strongly agree.

Table 1. Survey results from students on Utkus

Score / Assesment question	1	2	3	4	5
1) The usefulness of this system in explaining the meaning of a phrase?	3	3	3	4	2
2) The representativeness of the generated stages.	3	1	7	3	1
3) Overall satisfaction with the system.	2	4	3	6	0

It seems that volunteers have found the Utkus system interesting, but the current implementation does not completely satisfy their expectations. Most generated stages convey the meaning of the input sentence, but the final quality is lower than expected. Results of this user study can be explained by the following implementation details:

1. Incomplete depiction rules set (about 3 rules at the present moment) cannot cover all the actors actions;
2. Icons from The Noun Project are monochrome and too minimalistic, so they can't impress the end user;
3. The current Link Grammar for Russian implementation can process only one sentence at a time;
4. Spell checking and correction are not performed by the TTP system, therefore it shows an empty picture when the user input can't be understood on any processing stage;
5. Thesaurus that is used by the system was composed at the beginning of the 20th century and is quite incomplete to match the needs of today's users.

7. CONCLUSION

The described TTP system is intended for the Russian language processing and operates with the natural language analysis subsystem, the stage

processing subsystem, and the rendering subsystem. Every processing stage has been detailed and the basic design ideas of the text-to-picture synthesis system architecture have been highlighted.

TTP systems operate with a large amount of heterogeneous but linked and predefined information resources such as thesaurus, graphics gallery, ontology, and depiction rules. These rules specify actions of the input text actors and allow making the final picture more user-friendly and intuitive.

This approach has been examined at the Utkus general purpose text-to-picture synthesis system, and user study has been performed.

7.1. Further Work

Several reasons for the future work are available:

1. To enhance the linguistic analysis subsystem to handle such parts of speech as adjectives, pronouns, numerals, etc;
2. To solve the problem of ambiguity when generating the semantic representation;
3. To expand the supported types of depictors;
4. To produce graphical stubs for words that are unknown to system;
5. To improve the parser in purpose to handle texts of more than one sentence.

8. ACKNOWLEDGMENTS

The author would like to thank the Institute of Mathematics and Mechanics UrB RAS for the provided computer equipment. This work is partially supported by the U.M.N.I.K. program.

REFERENCES

1. **Yoshii, M., Flaitz, J.**: Second language incidental vocabulary retention: The effect of text and picture annotation types. CALICO journal 20(1), pp. 33–58 (2002).
2. **Åkerberg, O., Svensson, H., Schulz, B., Nugues, P.**: CarSim: an automatic 3D text-to-scene conversion system applied to road accident reports. In: Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics–Volume 2. pp. 191–194. Association for Computational Linguistics (2003).
3. **Goldberg, A., Rosin, J., Zhu, X., Dyer, C.**: Toward text-to-picture synthesis. In: NIPS 2009 Mini-Symposia on Assistive Machine Learning for People with Disabilities (2009).
4. **Li, H., Tang, J, Li, G., Chua, T.**: Word2image: Towards Visual Interpreting of Words. In: Proceedings of the 16th ACM international conference on Multimedia. pp. 813–816. ACM (2008).
5. **Zhu, X., Goldberg, A., Eldawy, M., Dyer, C., Strock, B.**: A text-to-picture synthesis system for augmenting communication. In: Proceedings of the National

Conference on Artificial Intelligence. vol. 22, p. 1590. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2007).
6. **Coyne, B., Sproat, R.**: WordsEye: an automatic text-to-scene conversion system. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. pp. 487–496. ACM (2001).
7. **Yamada, A., Yamamoto, T., Ikeda, H., Nishida, T., Doshita, S.**: Reconstructing spatial image from natural language texts. In: Proceedings of the 14th Conference on Computational Linguistics–Volume 4. pp. 1279–1283. Association for Computational Linguistics (1992).
8. **Adorni, G., Di Manzo, M., Giunchiglia, F.**: Natural Language driven Image Generation. In: Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics. pp. 495–500. Association for Computational Linguistics (1984).
9. ИА REGNUM. Петербургские депутаты разберутся с зарплатой логопедов. // http://www.regnum.ru/news/1511397.html
10. **Ustalov, D., Kudryavtsev, A.**: An Ontology-Based Approach to Text-to-Picture Synthesis Systems. In: Proceedings of the 2nd International Workshop on Concept Discovery in Unstructured Data (CDUD 2012). pp. 94–101.
11. **Ustalov, D.**: Distributed Dictionary-Based Morphological Analysis Framework for Russian Language Processing. In: Book of Abstracts of the International Young Scientists Conference "High Performance Computing and Simulation". pp. 82–83. 2012.
12. **Протасов, С.**: Преимущества грамматики связей для русского языка // Международная конференция Диалог 2005.